

# How to Easily Calculate AIC for Regression Models in Python

Authored by  
**stats writer**

December 5, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Calculate AIC for Regression Models in Python*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=105527>

The Akaike Information Criterion (AIC) is a fundamental metric used in statistics and machine learning to compare the quality of multiple competing regression models. It provides a standardized measure of a model's goodness-of-fit relative to other potential models, while simultaneously penalizing the model for excessive complexity. This balance between explanatory power and parsimony is crucial for developing models that generalize well to new data.

In data science workflows utilizing Python, the AIC of a statistical model is efficiently calculated using the specialized tools available in the statsmodels library. The calculation of the AIC depends on two primary factors: the number of independent variables (parameters) utilized to fit the model and the log-likelihood of the data given the model specification. A core principle of model selection using this criterion is that the lower the resulting AIC value, the better the model is considered to fit the data without resorting to unnecessary parameter proliferation.

This guide delves into the theoretical underpinnings of the AIC and provides a detailed, practical demonstration of how to implement and interpret AIC calculations when comparing candidate regression models in the Python environment, ensuring robust and evidence-based model selection.

## 1. Understanding the Akaike Information Criterion (AIC)

The challenge in statistical modeling is avoiding overfitting--the phenomenon where a model fits the training data almost perfectly but fails to generalize to new data because it has essentially memorized the noise. The AIC, developed by statistician Hirotugu Akaike, addresses this by providing a mechanism to estimate the predictive capability of the model based on relative information loss. It balances model complexity (the penalty term) against how well the model explains the data (the likelihood term).

Unlike simple metrics like R-squared, which always increases as more predictors are added, the AIC provides an objective measure of relative quality. When comparing different models fitted to the exact same dataset, the AIC estimates the expected Kullback-Leibler information loss. Therefore, it is essential to remember that AIC is useful only for relative comparisons; the absolute value of AIC has no inherent meaning, but the differences between AIC scores guide the selection process.

A difference in AIC scores of less than 2 generally suggests that both models are essentially equivalent in quality. Differences between 3 and 7 indicate that the model with the lower AIC is substantially better, and differences greater than 10 suggest that the model with the higher AIC is implausible and should be discarded. This tiered approach to interpretation ensures that model selection is not based on negligible differences.

## 2. The Mathematical Formulation of AIC

To fully grasp its application, it is beneficial to examine the mathematical definition of the AIC. The criterion is designed to quantify the trade-off between bias and variance, where simpler models suffer from higher bias and more complex models suffer from higher variance.

The Akaike information criterion (AIC) is a metric that is used to compare the fit of different regression models.

It is calculated as:

$$\text{AIC} = 2K - 2\ln(L)$$

where:

**K:** The number of model parameters. This count includes all estimated coefficients, such as the intercept, plus the estimate of the error variance. For a standard linear regression model, the default value of K is 2 (for the intercept and the variance term). Consequently, a model with one predictor variable will have a K value of  $2+1 = 3$ .

**$\ln(L)$ :** The log-likelihood of the model. This value reflects how likely the observed set of data is, assuming the proposed statistical model is the true mechanism generating the data. A higher log-likelihood indicates a better fit.

The AIC is expertly designed to find the model that explains the most observed variation in the data (maximizing the log-likelihood term), while simultaneously penalizing models that utilize an excessive number of parameters ( $2K$ ), thereby discouraging overfitting and promoting parsimony.

## 3. Calculating AIC using the Python statsmodels Library

For those working in Python, the calculation of the AIC is highly streamlined, requiring no manual implementation of the formula. The widely used statsmodels library, which offers extensive classes for statistical modeling, handles the derivation automatically.

When you fit a model using the Ordinary Least Squares (OLS) function from ``statsmodels``, the resulting fitted object contains all the necessary statistics, including the maximized log-likelihood and the parameter count. These components are then combined internally to yield the final AIC score, accessible via the built-in ``.aic`` attribute of the fitted model object.

This automated calculation ensures consistency and accuracy, allowing the data scientist to focus entirely on the selection process and the interpretation of the results, rather than the arithmetic. The following sections will use this methodology to compare two different model structures.

## 4. Example: Setting Up the Analysis with the mtcars Dataset

To illustrate the practical application of the AIC, we will use the classic **mtcars** dataset. Our objective is to determine the optimal set of predictors for modeling fuel efficiency, represented by the variable ``mpg`` (miles per gallon).

We begin by importing the necessary libraries--specifically ``pandas`` for data handling, and ``statsmodels.api`` for the regression analysis--and loading the data from a remote source. This initial preparation is crucial for defining our response and predictor variables accurately.

First, we'll load this dataset:

```
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
import pandas as pd

#define URL where dataset is located
url = "https://raw.githubusercontent.com/arabpsychology/Python-Guides/main/mtcars.csv"

#read in data
data = pd.read_csv(url)

#view head of data
data.head()

model mpg cyl disp hp drat wt  qsec vs am gear carb
0 Mazda RX4      21.0  6  160.0 110  3.90  2.620 16.46 0  1  4  4
1 Mazda RX4 Wag  21.0  6  160.0 110  3.90  2.875 17.02 0  1  4  4
2 Datsun 710     22.8  4  108.0  93  3.85  2.320 18.61 1  1  4  1
3 Hornet 4 Drive 21.4  6  258.0 110  3.08  3.215 19.44 1  0  3  1
4 Hornet Sportabout 18.7  8  360.0 175  3.15  3.440 17.02 0  0  3  2
```

We define two competing models, differing in their complexity (number of parameters), to test the efficacy of the AIC criterion:

Predictor variables in **Model 1**: ``disp`` (engine displacement), ``hp`` (gross horsepower), ``wt`` (weight), and ``qsec`` (1/4 mile time). This is a four-predictor model.

Predictor variables in **Model 2**: ``disp`` and ``qsec``. This is a two-predictor model.

## 5. Model 1: Fitting the Complex Regression Model

Model 1, the more complex of the two, incorporates four distinct predictor variables. Because we

are using OLS regression, we must always add a constant term to the predictor matrix to ensure the model includes an intercept. This means  $K=5$  (four predictors + intercept + error variance estimate, though the library calculates  $K$  implicitly). The greater number of parameters means Model 1 incurs a larger penalty term in the AIC calculation.

The following code snippet executes the fitting process and immediately retrieves the AIC value associated with this specification:

```
#define response variable
```

```
y = data
```

```
#define predictor variables
```

```
x = data]
```

```
#add constant to predictor variables
```

```
x = sm.add_constant(x)
```

```
#fit regression model
```

```
model = sm.OLS(y, x).fit()
```

```
#view AIC of model
```

```
print(model.aic)
```

```
157.06960941462438
```

The AIC value calculated for Model 1 is approximately **157.07**. This low value suggests a relatively good fit, but its true utility is realized only when compared side-by-side with the score of the competing model. If the four predictors collectively contribute significant explanatory power, the increased log-likelihood will overcome the penalty for the extra parameters.

## 6. Model 2: Fitting the Parsimonious Regression Model

Model 2 is intentionally designed to be simpler, using only two predictors (`disp`` and `qsec``). With the addition of the intercept, this model uses fewer parameters, resulting in a lower complexity penalty term ( $2K$ ) compared to Model 1. However, the critical question is whether this simplification sacrifices too much explanatory power, leading to a much higher AIC score.

We execute the same fitting procedure in Python, simply adjusting the definition of the predictor matrix `x``:

```
#define response variable
```

```
y = data
```

```
#define predictor variables
```

```
x = data]
```

```
#add constant to predictor variables
```

```
x = sm.add_constant(x)
```

```
#fit regression model
```

```
model = sm.OLS(y, x).fit()
```

```
#view AIC of model
```

```
print(model.aic)
```

```
169.84184864154588
```

The AIC of this second model turns out to be **169.84**. This value is significantly higher than the AIC achieved by Model 1, which immediately suggests that the complexity penalty reduction was not worth the resulting loss in model fit.

## 7. Comparing Results and Selecting the Optimal Model

The final step in the AIC process is the direct comparison of the scores. We have the following results:

**Model 1 AIC: 157.07** (Four predictors: disp, hp, wt, qsec)

**Model 2 AIC: 169.84** (Two predictors: disp, qsec)

Since the first model yields an AIC value that is considerably lower (by more than 12 points) than the second model, we conclude that **Model 1 is the better fitting model**. The variables excluded in Model 2 (`hp` and `wt`) must have provided substantial, necessary explanatory power. The increase in Model 1's fit was so significant that it easily compensated for the higher penalty associated with using more parameters.

This result validates the use of the AIC for objective model selection. Once Model 1 has been identified as the optimal structure, analysts can proceed with confidence to detailed interpretation. This includes analyzing the beta coefficients to understand the specific impact of each predictor and examining the R-squared value and residual plots to fully characterize the chosen regression model's performance and statistical validity.